Project Requirements Document (PRD)

Project Title:
Web-Based Platform for Professional Recruitment and Service Matching

1. Introduction

1.1 Purpose

The purpose of this project is to develop a web-based platform that connects employers with skilled professionals. The system will allow employers to post job opportunities and professionals to create profiles and apply for relevant positions.

1.2 Scope

The platform will provide:

- User registration and authentication
- Professional profile management
- Employer job posting and management
- Job search and application system
- Application tracking

The system will be developed using:

- Backend: Spring Boot
- Database: MySQL
- Version Control: GitHub
- Project Management: Jira
- API Testing: Postman

---

2. Stakeholders

Stakeholder	Description
Professionals	Users seeking jobs or services
Employers	Users posting job opportunities
Admin	System manager (optional)
Developer	System designer and implementer

---

3. User Roles

3.1 Professional (Job Seeker)

- Creates profile

- Searches jobs
- Applies for jobs
- Tracks application status

3.2 Employer

- Creates company profile
- Posts jobs
- Manages job listings
- Reviews applicants

3.3 Admin (Optional)

- Manages users
- Monitors system activity

4. Functional Requirements

4.1 Authentication & Authorization

The system should allow:

- Users to register using email and password
- Users to log in securely
- The system to validate user credentials
- Users to log out
- Role-based access (Professional / Employer / Admin)

4.2 Professional Module

The professional should be able to:

- Create a professional profile
- Update personal information
- Add skills and experience
- Upload resume (optional)
- View available jobs
- Search jobs by keyword or category
- Apply for a job
- View applied jobs
- Check application status

4.3 Employer Module

The employer should be able to:

- Create a company profile
- Update company information
- Post a new job
- Edit job details
- Delete a job
- View all posted jobs
- View applicants for a job
- Accept or reject applications

4.4 Job Management

The system should allow:

- Storage of job details
- Display of available jobs
- Filtering/search functionality
- Viewing job details

Job fields may include:

- Title
- Description
- Required skills
- Location
- Salary (optional)
- Job type

4.5 Application Management

The system should allow:

- Professionals to submit applications
- Employers to view applicant list
- Employers to update application status:
    - Pending
    - Accepted
    - Rejected

4.6 Admin Module (Optional – Phase 2)

The admin should be able to:

- View all users
- Deactivate user accounts
- Monitor job postings

5. Non-Functional Requirements

5.1 Security

- Password encryption
- JWT-based authentication
- Role-based access control

5.2 Performance

- API responses should be fast
- System should support multiple users

5.3 Reliability

- Proper error handling
- Validation of input data

5.4 Scalability

- Layered architecture
- Modular design

---

6. System Architecture

Layers:

- Controller Layer (REST APIs)
- Service Layer (Business Logic)
- Repository Layer (Database Access)
- Entity Layer (Data Models)

---

7. Database Overview (High-Level)

Entities

User

- id
- name

- email
- password
- role

ProfessionalProfile

- id
- user_id
- skills
- experience
- resume

EmployerProfile

- id
- user_id
- company_name
- company_description

Job

- id
- employer_id
- title
- description
- skills_required
- location

Application

- id
- job_id
- professional_id
- status

8. API Requirements

The system should provide REST APIs for:

- Authentication
- Profile management
- Job management
- Application management

All APIs should be tested using Postman and documented using Swagger.

9. Agile Development Approach

The project will follow Scrum methodology:

- Requirements converted into User Stories
- Managed in Jira Software
- Development in weekly sprints
- Version control using GitHub

10. Future Enhancements

- Email notifications
- Advanced job matching
- Frontend using React
- Docker deployment
- Admin dashboard
- Messaging between employer and professional

11. Sample User Stories (from this PRD)

Example conversions:

Requirement:
The professional should be able to apply for a job.

User Story:
As a professional, I want to apply for a job so that I can get hired.

Requirement:
The employer should be able to view applicants.

User Story:
As an employer, I want to view applicants so that I can select candidates.